OIPE IAP13
MAR 03 2006
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of	:	**Confirmation No. 9277**
Tokuo NAKATANI et al.	:	Attorney Docket No. 2001_0908A
Serial No. 09/891,176	:	Group Art Unit 2616
Filed June 26, 2001	:	Examiner Mishawn N. Dunn
DIGITAL VIDEO RECORDING APPARATUS	:	**Mail Stop: AMENDMENT**

SUBMISSION OF VERIFIED ENGLISH TRANSLATIONS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Verified English translations of Applicants' priority documents (i.e., JP 2000-190890 and JP 2000-190891) are submitted herewith.

Respectfully submitted,

Tokuo NAKATANI et al.

By: Kenneth W. Fields
Kenneth W. Fields
Registration No. 52,430
Attorney for Applicants

KWF/dib
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
March 3, 2006

THE COMMISSIONER IS AUTHORIZED TO CHARGE ANY DEFICIENCY IN THE FEES FOR THIS PAPER TO DEPOSIT ACCOUNT NO. 23-0975

VERIFICATION OF TRANSLATION

I, Kyozo Omori, translator of 831-9, Ono, Sanda, Hyogo, Japan, hereby declare that I am conversant with the English and Japanese languages and am a competent translator thereof. I further declare that to the best of my knowledge and belief the following is a true and correct translation made by me of Japanese Patent Application No. 2000-190890 filed on June 26, 2000.

Date: January 25, 2006

Kyozo Omori

KYOZO OMORI

[DOCUMENT] Patent Application

[OUR REFERENCE NO.] 2022520265

[FILING DATE OF APPLICATION] June 26, 2000

[To] Commissioner, Patent Office

[INTERNATIONAL PATENT CLASSIFICATION] G06F 7/00

G11B 7/00

[INVENTOR]

[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.

1006, Kadoma, Kadoma-City, Osaka

[NAME] Tokuo NAKATANI

[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.

1006, Kadoma, Kadoma-City, Osaka

[NAME] Kazuhiko NAKAMURA

[APPLICANT]

[CODE NO.] 000005821

[NAME] Matsushita Electric Industrial Co., Ltd.

[PATENT AGENT]

[CODE NO.] 100097445

[PATENT ATTORNEY]

[NAME] Fumio IWAHASHI

[APPOINTED PATENT AGENT]

[CODE NO.] 100103355

[PATENT ATTORNEY]

[NAME] Tomoyasu SAKAGUCHI

[APPOINTED PATENT AGENT]

[CODE NO.] 100109667

[PATENT ATTORNEY]

[NAME] Hiroki NAITO

[CHARGES]

[RECEIPT NO.] 011305

[AMOUNT] ¥21,000

[LIST OF ENCLOSURES]

Specification 1

Drawings 1

Abstract 1

[POWER OF ATTORNEY NO.] 9809938

[DOCUMENT] Specification

[TITLE OF THE INVENTION] Recorder

[CLAIMS]

[CLAIM 1] A recorder for use in a system for recording video data, which is generated by compressing and encoding a video signal, onto a disc, and at the same time, generating management information for managing the video data, wherein

if a recording of the video data onto the disc stops due to a system abnormality, the recorder performs a coordination process such that a size of the video data recorded on the disc becomes equal to a size of video data calculated from the management information.

[CLAIM 2] The recorder of CLAIM 1, wherein

the system abnormality is that the disc runs out of free space during the recording.

[CLAIM 3] The recorder of CLAIM 1, wherein

the system abnormality is that a power failure occurs during the recording.

[CLAIM 4] The recorder of CLAIM 1, wherein

as the coordination process that is performed such that the size of the video data recorded on the disc becomes equal to the size of video data calculated from the management information,

a last portion of the video data is deleted by a size that corresponds to the size of video data calculated from the management information.

[CLAIM 5] The recorder of CLAIM 1, wherein

as the coordination process that is performed such that the size of the video data recorded on the disc becomes equal to the size of video data calculated from the management information,

a management table is corrected such that the size of video data calculated from the management information becomes equal to the size of the video data.

[DETAILED DESCRIPTION OF THE INVENTION]

0001

[FIELD OF THE INVENTION]

The present invention relates to a technology for recording digital data of video onto a recording medium.

0002

[DESCRIPTION OF THE RELATED ART]

In the history of establishing the technology for recording video data, initially, tape mediums were mainly used as the recording mediums. As it became possible to record video data, a function to perform special playbacks (fastforward, rewinding and the like) was provided. Such a function has become commonplace nowadays. When data is recorded onto a tape medium, the video data is consecutively recorded onto the tape. As a result, the recording order of video data on the tape is used as the playback order as it is, and a special playback is achieved by performing an intermittent playback while physically fastforwarding or rewinding the tape.

0003

And then optical discs such as CD were developed and came

to practical use as mediums for recording data thereon. Disc mediums are superior than the tape mediums in accessibility. When tape is used, it is necessary to move the tape to a point where desired data is recorded. For this purpose, the tape is moved one-dimensionally, which takes much time.

0004

However, in the case of disc mediums, a two-dimensional moving process is performed. That is to say, while the disc is moved, the pickup is moved to read data. Due to this construction, disc mediums have remarkably higher accessibility than the tape mediums. To fully make use of the accessibility, management information is required to manage where on the disc the data is recorded. When disc mediums first appeared, they did not attract much attention as the mediums for recording video data thereon due to their small capacity (among them, the video CD has been put into practical use, but not so popular). It is considered that the unpopularness of the medium is also attributable to the limitation "read-only".

0005

In recent years, however, DVD-RAM, which is a phase-change optical disc having several giga bytes of capacity, appeared. Coupled with the practical use of MPEG (MPEG2), which is an encoding standard for digital audio-visual data (hereinafter, AV data), the DVD-RAM is being used as an audio-visual recording/playback medium, as well as being used for computers. As the specifications of information required to achieve the special playbacks and the like on the video data recorded on the DVD-RAM, DVD Specifications for Rewritable/Re-recordable Discs were established and issued. This completes the technologies required for recording video data onto

recording mediums.

0006

(Explanation of MPEG)

The AV data recorded on the DVD-RAM needs to conform to an international standard called MPEG (ISO/IEC13818).

0007

Although DVD-RAM has a large capacity of several giga bytes, it is not enough to record non-compressed digital AV data as it is. This necessitates the use of a method of compressing AV data and recording the compressed AV data. As the AV data compression method, MPEG (ISO/IEC13818) has become widespread. Owing to the recent improvement in the LSI technology, MPEG codec (expansion/compression LSI) has come to practical use. This has enabled DVD recorders to perform MPEG expansion/compression.

0008

MPEG has the following two main characteristics for realizing highly-efficient data compression. One characteristic is that its moving-image data compression adopts a compression method using the temporal correlation between frames, as well as a conventional compression method using the space frequency. For data compression, MPEG classifies each frame (also referred to as "picture" in MPEG) into three types: I-picture (Intra-coded picture); P-picture (Predictive picture --- a picture using the intra-coding and references from the past); B-picture (Bidirectionally-predictive picture --- a picture using the intra-coding and references from the past and future).

0009

Fig. 1 shows relationships among I-, P-, and B-pictures. As

shown in Fig. 1, a P-picture refers to an I- or P-picture that is closest thereto in the past, and a B-picture refers to an I- or P-picture that is closest thereto in the past and future. Also, since, as shown in Fig. 1, a B-picture refers to an I- or P-picture in the future, there may occur a case where the display order of the pictures does not match the coding order thereof.

0010

To realize, in a playback of data stored in a storage medium, trick plays such as fastforward, rewinding, and a playback from a middle position, MPEG defines a structure called GOP (Group Of Pictures). More specifically, a GOP is composed of a set of frames that includes at least one I-picture so that a random access can be performed in units of GOPs. This enables trick plays to be realized by performing a skip playback, playing back only the I-pictures in GOPs.

0011

The second characteristic of MPEG is that it can dynamically assign amounts of coding in units of pictures in accordance with the complexity level of the images. An MPEG decoder includes an input buffer. It is possible to assign a large amount of coding to a complex image that is difficult to compress, by preliminarily storing data in the decoder buffer.

0012

The audio data used in DVD-RAM can be selected from three types: MEPG audio using compressed data; Dolby digital (AC-3) using compressed data; and LPCM using non-compressed data. The Dolby digital and the LPCM uses a fixed bit rate. For the MEPG audio, it is possible to select a size from several sizes in units of audio frames, though it is not so large as a video stream.

0013

The AV data described above is multiplexed into one stream by a method called MPEG system. Fig. 2 shows the construction of an MPEG system. The element 21 is a pack header, 22 a packet header, and 23 a payload. The MPEG system has a hierarchical structure composed of packs and packets. Each packet includes a packet header 22 and a payload 23. The AV data is divided into pieces having an appropriate size, and the divided pieces of AV data are stored into the payloads 23, respectively.

0014

Recorded in the packet header 22 are, as information of the AV data stored in the payload 23: stream ID for identifying the stored data; DTS (Decoding Time Stamp) indicating the decoding time of the data in the payload, in the accuracy of 90 kHz; and PTS (Presentation Time Stamp) indicating the presentation time of the data in the accuracy of 90 kHz (it should be noted here that the DTS is omitted when, as is the case with audio data, decoding and presentation are performed at the same time). Each pack is composed of a plurality of packets. In the case of the DVD-RAM, each packet is used as a pack. And accordingly, each pack is composed of a pack header 21 and a packet (composed of a packet header 22 and a payload 23). Recorded in the packet header 22 is SCR (System Clock Reference) that indicates the time at which the data in the pack is input into the decoder buffer, in the accuracy of 27 MHz. In DVD-RAM, the MPEG system stream as described above is recorded on presumption that one pack corresponds to one sector (= 2,048 bytes).

0015

The following describes the decoder for decoding the

above-described MPEG system stream. Fig. 3 is a block diagram showing a decoder model (P-STD) of the MPEG system decoder. The element 31 is an STC (System Time Clock) for providing a standard time in the decoder, 32 is a demultiplexer for decoding, namely, demultiplexing the system stream, 33 is a video buffer for a video decoder, 34 is the video decoder, 35 is a reorder buffer for temporarily storing I- and P-pictures to absorb the difference between the data order and the display order that occurs with the I-, P-, and B-pictures, 36 is a switch for adjusting the output order of the I-, P-, and B-pictures stored in the reorder buffer, 37 is an audio buffer for an audio decoder, and 38 is the audio decoder.

0016

The MPEG system decoder processes the MPEG system stream as follows. The demultiplexer 32 inputs a pack when the time provided by the STC 31 matches the SCR written in the pack header of the pack. The demultiplexer 32 decodes the stream ID in the packet header, transfers the payload data to the decode buffers for each stream, and extracts the PTS and DTS from the packet header. The video decoder 34 extracts the picture data from the video buffer 33 at a time when the time provided by the STC 31 matches the DTS, performs the decoding process, stores the I- and P-pictures into the reorder buffer 35, and outputs the B-pictures for display. When the video decoder 34 is decoding the I- and P-pictures, the switch 36 is set to output the I- or P-pictures to the reorder buffer 35, and when the video decoder 34 is decoding the B-pictures, the switch 36 is set toward the video decoder 34. The audio decoder 38, as is the case with the video decoder 34, extracts one audio frame of data from the audio buffer 37 at a time when the time provided by the STC 31 matches the

PTS (there is no DTS for audio), and performs the decoding process.

0017

Next, the MPEG system stream multiplexing method will be described with reference to Fig. 4. In Fig. 4, (a) indicates video frames, (b) a video buffer, (c) an MPEG system stream, and (d) audio data. The horizontal axis represents a time axis that is common to (a) through (d). In (b), the vertical axis represents the amount of buffer occupation (amount of data stored in the video buffer), and the thick line indicates the temporal transition of the amount of buffer occupation. The slant of the thick line corresponds to the video bit rate, indicating that the data is input into the buffer at a constant rate. Also, (b) shows that the amount of buffer occupation decreases at regular intervals. This indicate that the data is decoded. Further, the points at the intersections of slant dotted lines and the time axis indicate the times at which the data of the video frames starts to be transferred to the video buffer.

0018

Now, an explanation using a complex image A in the video data will be given. As shown in (b) of Fig. 4, since the image A requires a large amount of coding, the data transfer to the video buffer is started at time t1 prior to the decoding time of the image A. (The time period from the data input start time t1 to the decoding time is referred to as vbv_delay) For this reason, the data, as the AV data, starts to be multiplexed at the position (time) of the video pack indicated by the arrow with hatching. On the other hand, in the case of audio data that does not require the dynamic control of the amount of coding as video data, there is no need to start transferring data much earlier than the decoding time, and generally the audio

data is multiplexed a little before the decoding time.

0019

Therefore, in terms of video data and audio data that are played back at the same time, the video data is started to be multiplexed earlier than the audio data. It should be noted here that MPEG defines that the time period for which data can be stored in the buffer is limited, and that all the data except for still picture data should be output from the buffer to the decoder within one second after the data is input into the buffer. Accordingly, the difference between video data and audio data in multiplexing is one second at the maximum (more accurately, the time for reordering of the video data may be added to the difference).

0020

In the above-described example, video precedes audio. However, theologically, audio may precede video. Such data can be intentionally generated by, for example, preparing a simple image with high compression rate for the video data and unnecessarily transferring the audio data. It should be noted here that the time period allowed for the preceding is one second at the maximum due to the limitation by MPEG.

0021

(Logical Structure of DVD-RAM)

Here, the logical structure of DVD-RAM will be described. The portion (a) of Fig. 5 shows the data structure of the file system in the disc, and (b) shows the physical sector addresses in the disc. The lead-in area is provided at the start of the physical sector addresses. The lead-in area stores a standard signal necessary for stabilizing the servo, an identification signal against other mediums,

and the like. The lead-in area is followed by the data area in which logically effective data is recorded. The physical sector addresses has the lead-out area at the end. The lead-out area stores, as the lead-in area, a standard signal and the like.

0022

Recorded at the start of the data area is management information for the file system, the management information called volume information. As shown in (a) of Fig. 5, use of the file system enables data in the disc to be treated as a directory or a file.

0023

In the structure defined in the VIDEO RECORDING standard, as shown in (b) of Fig. 5, the management information is present under the DVD_RTAV directory that is immediately under the ROOT directory. Under the DVD_RTAV directory, there are a management information file "VR_MANGR.IFO" (hereinafter referred to as IFO file) and a plurality of (at least one) AV files. The AV file is classified into three files: moving image (VR_MOVIE.VRO); still image (VR_STILL.VRO); and audio dubbed for the still image (VR_AUDIO.VRO). One IFO file is provided as information for managing the three AV files.

0024

(Explanation of Management Information File in VIDEO RECORDING Standard)

Here, the structure of the IFO file defined in the VIDEO RECORDING standard will be explained. This explanation centers on the management information for video, and in particular centers on PG (explanation of the part that is irrelevant to the present patent application is omitted).

0025

As shown in Fig. 6, the IFO file includes three video-related tables: VOB_STI (VOB stream attribute information) table; VOBI (VOB information) table; and PGCI (PGC information) table. The VOB is an MPEG program stream. The Cell is a logical playback unit that refers to a given partial section (or the whole section) in the VOB. The PGC defines the playback order of the Cell. In the VIDEO RECORDING standard, strictly speaking, the VOB differs from the MPEG system stream. However, in this explanation, they are treated as the same one (The following is the difference. The system stream should end with the program end code. In contrast, there is no such definition in regards with VOBs in the VIDEO RECORDING standard).

0026

A VOB is a set of VOBUs. The VOBU is a data unit that includes an MPEG program stream generated by multiplexing one or more sets of GOPs of MPEG video data, its program stream, and a plurality of interleaved audio packs. It should be noted here that the GOPs included in one VOBU are complete themselves in the VOBU. Also, the playback time period of one VOBU is limited to a certain range, and the encoder needs to generate the VOBU to meet the limitation.

0027

The aforesaid VOBI in the IFO file is the management information of the above-described VOB. The VOBI table records therein the number of VOBIs (VOB_SRP_Ns) and VOBIs. The VOBI includes the type of the VOB (VOB_Type), playback start time (VOB_Start_PTM), playback end time (VOB_End_PTM), information relating to the time at which the start of the VOB is recorded (VOB_REC_TM), a reference pointer to a VOB STI (to be described later) that shows the attribute information of the VOB (VOB_STIN), and time map information (TMAPI).

0028

The TMAPI is management information used for a special playback or a jump playback, and includes information regarding a VOBU that constitutes a VOB. More specifically, the TMAPI includes TIME MAP GENERAL information (TMAP_GI), a time map entry (TM_ENT), and a VOBU entry (VOBU_ENT). As shown in the upper-left portion of Fig. 7, TMAP_GI includes a VOB address offset (ADE_OFS), a playback time offset from the start of the VOB of FIRST TM_ENT (TM_OFS), the number of time map entries (TM_ENT_Ns), and the number of VOBU entries (VOBU_ENT_Ns).

0029

Fig. 7 shows the TMAPI in relation to the stream. As the figure indicates, ADR_OFS is OFFSET from the stream file start to the VOB start on presumption that the stream file start is "0". TM_OFS is OFFSET from the playback start time of TM_ENT#1 on presumption that the VOB start time is "0". In general, TM_OFS is "0" immediately after recording. However, TM_OFS becomes a value other than "0" when, for example, the starting data of the VOB is deleted by editing. It should also be noted that the playback start time indicated by TM_ENT#j does not necessarily match the VOBU start time. This occurs since the VOBUs can have uneven playback time lengths. As a result, as shown in Fig. 7, TM_ENT has information called TM_DIFF as well as the VOBU number (VOBU_ENTN) to indicate a difference between the playback start time of the VOBU_ENT of the VOBU_ENTN and the playback start time of the TM_ENT itself. TM_ENT also has the start address (VOBU_ADR) of the VOBU it indicates. VOBU_ADR is an offset from the VOB start. By combining VOBU_ADR with ADR_OFS of TMAP_GI, it is possible to calculate an address from the start of the VR_MOVIE.VRO file, which makes it possible to directly access a given VOBU.

0030

VOB STI is attribute information of VOB being MPEG program stream. The information of VOB STI may be incorporated into each VOBI, but the standard provides the specification such that VOBs having a common attribute refer to the same VOB STI, thereby restricting the size of the IFO file.

0031

As shown in Fig. 6, the above-mentioned VOB STI table records therein the number of VOB STIs (VOB_STI_Ns) and VOB STIs. Each VOB STI includes the Video Attribute (video attribute information), the number of audio streams (Number of Audio Streams), the number of sub picture streams (Number of Sub Picture Streams), Audio Attribute (audio attribute information), Sub Picture Attribute (sub picture attribute information), and Sub Picture Color Pallet.

0032

The PGCI table, which is management information of PGC, records therein the number of PGCIs (PG_Ns) and the PGCIs. Each PGCI includes the number of CellIs that are present in the PGC (C_Ns), and includes CellIs that are management information of each Cell.

0033

Each CellI includes a search pointer to the VOBI of VOB corresponding to the Cell (VOBI_SRP), Cell playback start time (Cell_Start_PTM), Cell playback end time (Cell_End_PTM), the number of entry points for the Cell (EPI_Ns), and entry point information (Cell_EPI) table.

0034

[THE PROBLEMS THE INVENTION IS GOING TO SOLVE]

As described in the Description of the Related Art, the

VR_MOVIE.VRO files being MPEG streams are encoded in units of GOPs. The IFO file defined in the VIDEO RECORDING standard also has the time map information, as explained earlier. The VR_MOVIE.VRO files are managed in units of VOBUs that constitute each VOB. Managing the VR_MOVIE.VRO files in units of VOBUs makes it easy to perform trick plays and editing.

0035

When taken into consideration that the minimum management unit of the IFO file is VOBU and that the minimum structure unit of the VR_MOVIE.VRO file is VOBU, it is preferable that the VR_MOVIE.VRO files match the IFO files in units of VOBUs.

0036

For example, suppose that the end of a program that is last recorded is deleted by editing. In case more VOBUs than the VOBUs managed by the IFO file are recorded in the actual VR_MOVIE.VRO file, then if only the information of the IFO file is edited, data in the middle portion of the stream is deleted, and the data in the portion not managed by the IFO file remains undeleted in the VR_MOVIE.VRO file. Required to avoid this is a troublesome process in which the file size of the VR_MOVIE.VRO file is acquired from the file system, the start address of the VOBU to be deleted is subtracted from the acquired size, and the size of the deletion is determined.

0037

During an ordinary recording, the system stream is recorded into the VR_MOVIE.VRO file, while the recorded VOBU entries are registered with the IFO file. With such an operation, the IFO file always matches the stream.

0038

However, when the disc has barely enough capacity for the recording, the matching is not ensured. The disc medium does not always has free areas. There may be a case where an error sector is found after the write process is actually performed, or a case where data writing is not available due to contamination on the disc surface.

0039

For this reason, performing data writing after conforming that the disc has free areas does not necessarily ensure that the whole VOBU requested by the write request is written. Also, even if the VOBU information of the IFO file is updated only when the whole VOBU is written, if the VOBU is written partially, the VR_MOVIE.VRO file indicates that the VOBU has been written partially. This creates a mismatch between the VR_MOVIE.VRO file and the IFO file.

0040

There is a case that creates a mismatch between the VR_MOVIE.VRO file and the IFO file, other than the above-described case where the disc has barely enough capacity for the recording. It is a power failure that occurs during recording. If a power failure occurs while the stream data of the VOBU is written into the VR_MOVIE.VRO file, a complete writing of the data is not ensured, and there is no assurance that the requested size of data is written. When this happens, a mismatch occurs between the IFO file and the VR_MOVIE.VRO file.

0041

[MEANS TO SOLVE THE PROBLEMS]

To solve the above problems, CLAIM 1 provides a recorder for use in a system for recording video data, which is generated by compressing and encoding a video signal, onto a disc, and at the same time, generating management information for managing the video data,

wherein if a recording of the video data onto the disc stops due to a system abnormality, the recorder performs a coordination process such that a size of the video data recorded on the disc becomes equal to a size of video data calculated from the management information.

0042

CLAIM 2 provides the recorder of CLAIM 1, wherein the system abnormality is that the disc runs out of free space during the recording.

0043

CLAIM 3 provides the recorder of CLAIM 1, wherein the system abnormality is that a power failure occurs during the recording.

0044

CLAIM 4 provides the recorder of CLAIM 1, wherein as the coordination process that is performed such that the size of the video data recorded on the disc becomes equal to the size of video data calculated from the management information, a last portion of the video data is deleted by a size that corresponds to the size of video data calculated from the management information.

0045

CLAIM 5 provides the recorder of CLAIM 1, wherein as the coordination process that is performed such that the size of the video data recorded on the disc becomes equal to the size of video data calculated from the management information, a management table is corrected such that the size of video data calculated from the management information becomes equal to the size of the video data.

0046

[EMBODIMENTS OF THE INVENTION]

The present invention will be described in detail through a DVD recorder which is the first embodiment of the present invention.

First, the basic structure of the DVD recorder will be explained.

0047

(Block Diagram of DVD Recorder)

Fig. 8 is a block diagram showing the DVD recorder. In Fig. 8, the element 81 is a user interface unit for displaying for a user and receiving requests from the user, 82 is a system control unit for managing and controlling the whole DVD recorder and for generating stream management information, 83 is an input unit composed of a camera and a microphone or a TV tuner, 84 is an encoder unit composed of a video encoder VE, an audio encoder AE, and a system encoder SE, 85 is an output unit composed of a monitor and a speaker, 86 is a decoder unit composed of a system decoder, an audio decoder, and a video decoder, 87 is a truck buffer, 88 is a drive, and 89 is a time management unit for managing the time in the system.

0048

(Normal Recording Operation)

First, the recording operation of the DVD recorder will be explained with reference to Fig. 8. The user interface unit 81 receives a request from a user. The user interface unit 81 then conveys the user's request to the system control unit 82. The system control unit 82 analyzes the user's request and requests processes to each module. If the user's request is to record the video and audio, the system control unit 82 sets the encoder unit 84 to the settings requested by the user interface unit 81 (for example, how to compress video, or the system bit rate), generates forms of the VOB STI, VOBI, and CellI of the management information shown in Fig. 6, and requests the encoder unit 84 to encode the video frames and audio. In doing this, the system control unit 82 acquires the current time from the

time management unit 89, and sets the acquired current time to VOB_REC_TM in VOBI.

0049

The encoder unit 84 generates video data by video-encoding video frames sent from the input unit 83, and at the same time generates audio data by audio-encoding audio sent from the input unit 83. The generated video and audio data are system-encoded and formed into a system stream that is an MPEG program stream. The generated system stream is transmitted to the truck buffer 87. At the same time, each time the system-encoding of a VOBU is completed, the encoder unit 84 notifies the system control unit 82 of VOBU information concerning the VOBU that was completely encoded. The system control unit 82 updates the management information shown in Fig. 6 based on the VOBU information.

0050

The VOBU information is classified into the following:

- VOBU Start PTM (video frame playback start time in VOBU);
- Reference Picture Size (size of the first I-picture when the VOBU start is "0");
- VOBU Size (the number of multiplexed units);
- VOBU PB Time (playback time);
- Aspect ratio;
- AUDIO mode; and
- The number of AUDIO streams.

More specifically, the following processes are executed based on the above-listed information: update of TMAPI (addition of TMAP_ENT, VOBU_ENT); and update of VOB_End_PTM, Cell_End_PTM. The VOBU information that is received first after starting a recording is used

to set VOB_Start_PTM, Cell_Start_PTM.

0051

After a predetermined amount of system stream is stored in the truck buffer 87, the system control unit 82 records the system stream data stored in the truck buffer 87 to a DVD-RAM disc via the drive 88.

0052

The stop request from the user is conveyed to the system control unit 82 via the user interface unit 81. The system control unit 82 then sends a video/audio recording stop instruction to the encoder unit 84. The encoder unit 84 ends the encoding with the system-encoding of the audio frame that was generated immediately after it, transfers the generated system stream data to the truck buffer 87, and conveys the end of the encoding process to the system control unit 82. The system control unit 82 records all the remaining system stream data stored in the truck buffer 87 onto the DVD-RAM disc via the drive 88.

0053

After the above-described operation, the system control unit 82 records the aforesaid VOBI and CellI onto the DVD-RAM disc via the drive 88.

0054

(Occurrence of DISC FULL Halfway through Recording)

Now, a case where the disc becomes full during a recording will be explained, as an example of the case where a mismatch between the IFO file and the VR_MOVIE.VRO file occurs.

0055

As described earlier, each time the system-encoding of a VOBU

is completed, the encoder unit 84 transmits an encoding completion notification to the system control unit 82, and the VOBU stream data is transmitted to the truck buffer 87.

0056

The drive 88 writes the data stored in the truck buffer 87 onto the disc. Here, if the disc does not have enough free space, the drive 88 continues to write the data onto the disc until the disc becomes full, and then notifies the system control unit 82 of a write error.

0057

Upon receiving the notification of the write error, the system control unit 82 performs a process for coordinating the IFO file with the VR_MOVIE.VRO file on the disc. This is because the VOBU information is transmitted to the system control unit 82 on presumption that all the VOBUs are recorded, and if the writing ends halfway through, a mismatch occurs between the IFO file and the VR_MOVIE.VRO file.

0058

There is also a possibility that data remains in the truck buffer 87. For this reason, it is difficult for the system control unit 82 to determine up to which portion the VOBU notified from the encoder unit 84 has been recorded onto the disc.

0059

The above-described conditions taken into account, the system control unit 82 acquires the size of the VR_MOVIE.VRO file from the file system, compares it with the logical file size calculated from the IFO file, and performs the coordination process of the VR_MOVIE.VRO file and the IFO file according to the flowchart shown in Fig. 9. Fig. 9 will be explained later.

0060

It should be noted here that the encoder unit 84 may send the VOBU information to the system control unit 82 only in regards with VOBUs that have been written completely. However, in this case also, as explained in "THE PROBLEMS THE INVENTION IS GOING TO SOLVE", if an error occurs halfway through "VOBU WRITE", the last written VOBU needs to be deleted from the VR_MOVIE.VRO file.

0061

(Power Failure during Recording)

Now, a case where a power failure occurs during a recording will be explained, as another example of the case where a mismatch between the IFO file and the VR_MOVIE.VRO file occurs.

0062

In Fig. 8, as explained earlier, the VOBU stream data is temporarily stored in the truck buffer 87 during a recording. If a power failure occurs and the power supply stops under this condition, the drive 88 is forcibly stopped while it is recording stream data onto the disc. When this happens, the system control unit 82 cannot determine up to which portion the stream data, which has been requested by the WRITE request, has been recorded onto the disc. This necessitates the same process, which is performed when the disc becomes full during recording, to be performed.

0063

(Coordination Process between IFO File and VR_MOVIE.VRO File)

The coordination process between the IFO file and the VR_MOVIE.VRO file will be described with reference to Figs. 9 and 10.

0064

First, in step 901, the size of the VR_MOVIE.VRO file (hereinafter referred to as fs_size) is obtained from the file system. Next, in step 902, the size of the logical stream file (hereinafter referred to as tmp_size) is calculated from the VOBU table in the IFO file.

0065

In step 903, the fs_size is compared with the tmp_size. Here, if the fs_size is equal to the tmp_size, the process ends since it is judged that the files have already been coordinated with each other. If it is judged in step 903 that the fs_size is different from the tmp_size, the control moves to step 904 in which it is judged whether the fs_size is larger than the tmp_size.

0066

If it is judged that the fs_size is larger than the tmp_size, meaning that the VR_MOVIE.VRO file includes a VOBU that is not managed by the IFO file, the control moves to step 905 in which data having the size corresponding to tmp_size is deleted by fs_size - tmp_size from the end of the stream file.

0067

If it is judged that the tmp_size is larger than the fs_size, the control moves to step 906. Fig. 10 shows the details of the process performed in step 906.

0068

(When IFO File Has Invalid VOBU Information)

If the tmp_size, which is calculated from the IFO file, is larger than the fs_size, which is the real size of the VR_MOVIE.VRO file, the control goes to step 1001 in which the last VOBU entry is deleted from the VOBU map. In doing this, it is necessary to update

the following information shown in Fig. 6: VOB_End_PTM in VOBI; VOBU_ENT_Ns in TMAP GI; and Cell_End_PTM in CELLI.

0069

VOB_ENT_PTM, VOBU_ENT_Ns in TMAP GI, and Cell_End_PTM are updated each time the encoder unit 84 notifies the system control unit 82 of the VOBU information. VOB_End_PTM and Cell_End_PTM are updated based on the following expressions each time the VOBU information is notified.

0070

In the case of NTSC:

VOB#END#PTM (new) = VOB#END#PTM (old)
+ the number of frames of the added VOBU * 3003

CELL#END#PTM (new) = CELL#END#PTM (old)
+ the number of frames of the added VOBU * 3003

In the case of PAL:

VOB#END#PTM (new) = VOB#END#PTM (old)
+ the number of frames of the added VOBU * 3600

CELL#END#PTM (new) = CELL#END#PTM (old)
+ the number of frames of the added VOBU * 3600

After step 1001 in which a VOBU is deleted from the VOBU map, VOB_End_PTM and Cell_End_PTM are updated based on the following expressions in steps 1002 and 1003.

In the case of NTSC:

VOB#END#PTM (new) = VOB#END#PTM (old)
- the number of frames in VOBU to be deleted * 3003

CELL#END#PTM (new) = CELL#END#PTM (old)
- the number of frames in VOBU to be deleted * 3003

In the case of PAL:

VOB#END#PTM (new) = VOB#END#PTM (old)

- the number of frames in VOBU to be deleted * 3600

CELL#END#PTM (new) = CELL#END#PTM (old)

- the number of frames in VOBU to be deleted * 3600

In step 1003, VOBU_ENT_Ns is updated, namely, is decreased by "1".

0071

It should be noted here that as shown in Fig. 7, TM_ENTs refer to VOBU_ENTs. Accordingly, a deletion of VOBU_ENTs may create a TM_ENT that refers to a VOBU that does not exist. For this reason, it is necessary to check LAST TM_ENT each time VOBU_ENT is deleted. As a result, in step 1005, it is judged whether it is true that the VOBU_ENTN referred to by LAST TM_ENT is no larger than VOBU_ENT_Ns. If the VOBU_ENTN is larger than VOBU_ENT_Ns, meaning that there is no VOBU_ENT to refer to, the control moves to step 1006 in which LAST TM_ENT is deleted, and then the control moves to step 1007 in which TM_ENT_Ns in TMAP_GI is decreased by "1", and then the control moves to step 1008.

0072

If it is judged affirmatively in step 1005, the control goes to step 1008, skipping the process of TM_ENT. In step 1008, tmp_size is updated, namely, is decreased by the data size of the deleted VOBU. In step 1009, the fs_size is compared with the updated tmp_size. Here, if the fs_size is equal to the updated tmp_size, the coordination process ends since it is judged that the files are coordinated with each other.

0073

If it is judged in step 1009 that the fs_size is larger than

the tmp_size, meaning that a VOBU not managed by the IFO file is recorded in the VR_MOVIE.VRO file, the control goes to step 1011 in which as much stream data as tmp_size is deleted by fs_size - tmp_size from the end of the VR_MOVIE.VRO file. This completes coordinating the IFO file with the VR_MOVIE.VRO file, and the coordination process ends. If it is judged in step 1010 that the tmp_size is larger than the fs_size, meaning that the IFO file still have an invalid piece of VOBU information, the control goes to step 1001 to repeat the steps.

0074

[EFFECTS OF THE INVENTION]

The present invention coordinates the IFO file with the VR_MOVIE.VRO file. This prevents the IFO file from including an invalid piece of VOBU information, and avoids generating a disc that does not conform to the standard.

0075

Coordinating the IFO file with the VR_MOVIE.VRO file further allows such editing that uses only the information of the IFO file. This simplifies the editing (if the IFO file is not coordinated with the VR_MOVIE.VRO file, a VOBU that should be deleted remains undeleted when only the information of the IFO file is used in the editing).

[BRIEF DESCRIPTION OF THE DRAWINGS]

Fig. 1 shows relationships among pictures in the MPEG video stream.

Fig. 2 shows the construction of the MPEG system stream.

Fig. 3 is a block diagram showing the construction of the MPEG system decoder (P-STD).

Fig. 4:

(a) a figure showing video data;

(b) a figure showing video buffer;

(c) a figure showing MPEG system stream; and

(d) a figure showing audio data.

Fig. 5:

(a) a figure showing directory structure; and

(b) a figure showing a physical arrangement on the disc.

Fig. 6 shows management information data.

Fig. 7 shows relationships between the stream and the TIME MAP information.

Fig. 8 shows the construction of the DVD recorder.

Fig. 9 is a flowchart of the coordination process between the IFO file and the VR_MOVIE file (the first part).

Fig. 10 is a flowchart of the coordination process between the IFO file and the VR_MOVIE file (the second part).

[DESCRIPTION OF CHARACTERS]

21 pack header

22 packet header

23 payload

31 STC

32 demultiplexer

33 video buffer

34 video decoder

35 reorder buffer

36 switch

37 audio buffer

38 audio decoder

81 user interface unit

82 system control unit

83 input unit

84 encoder unit

85 output unit

86 decoder unit

87 truck buffer

88 drive

89 time management unit

[DOCUMENT] Abstract

[SUMMARY]

[AIM] To coordinate the management information with the stream file so as not to generate a disc that does not conform to the standard.

[MEANS TO ACHIEVE THE AIM] When the disc runs out of free space or a power failure occurs and a system abnormally ends during recording of a stream, the stream file size, which is obtained from the file system, is compared with the logical stream file size, which is calculated from the management information. If the actual stream file size is larger than the logical stream file size, unnecessary data is deleted; and if the logical stream file size is larger than the actual stream file size, unnecessary management data is deleted. In this way, a coordination between the stream file and the management information is achieved.

[SELECTED FIGURE] Fig. 10

FIG. 1

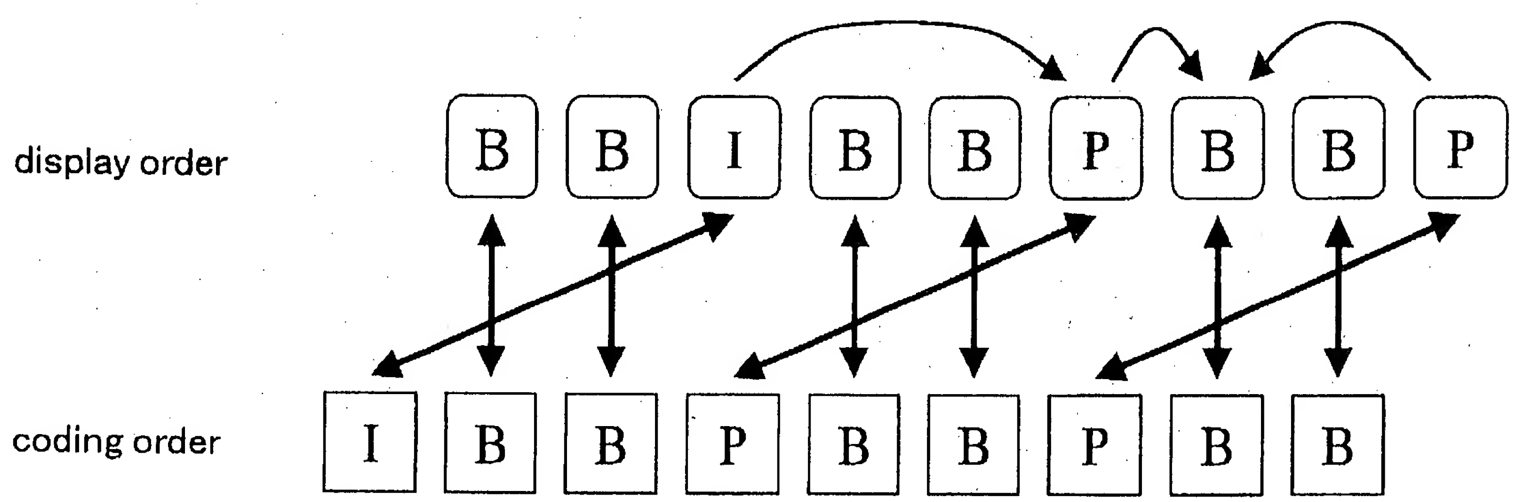


FIG. 2

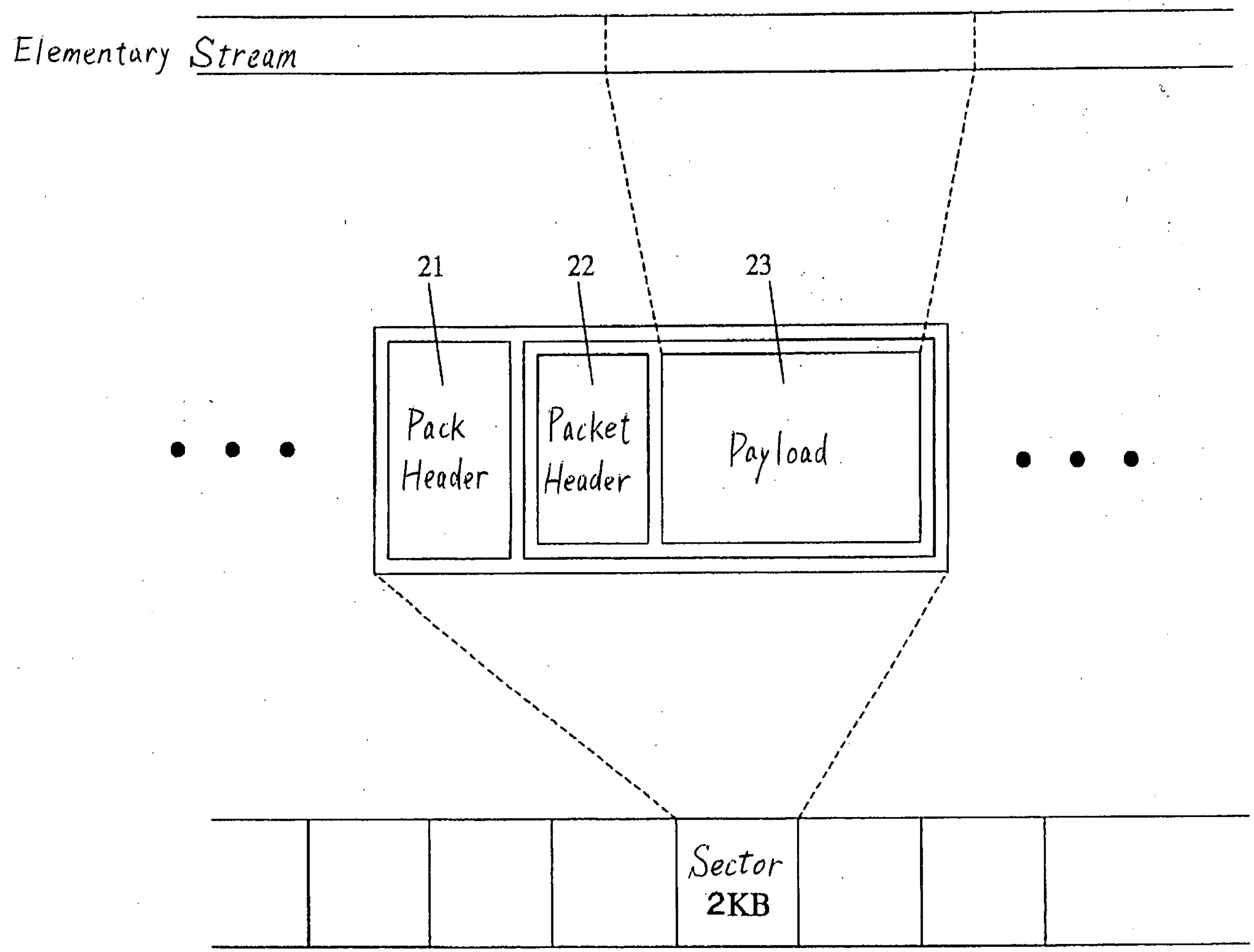
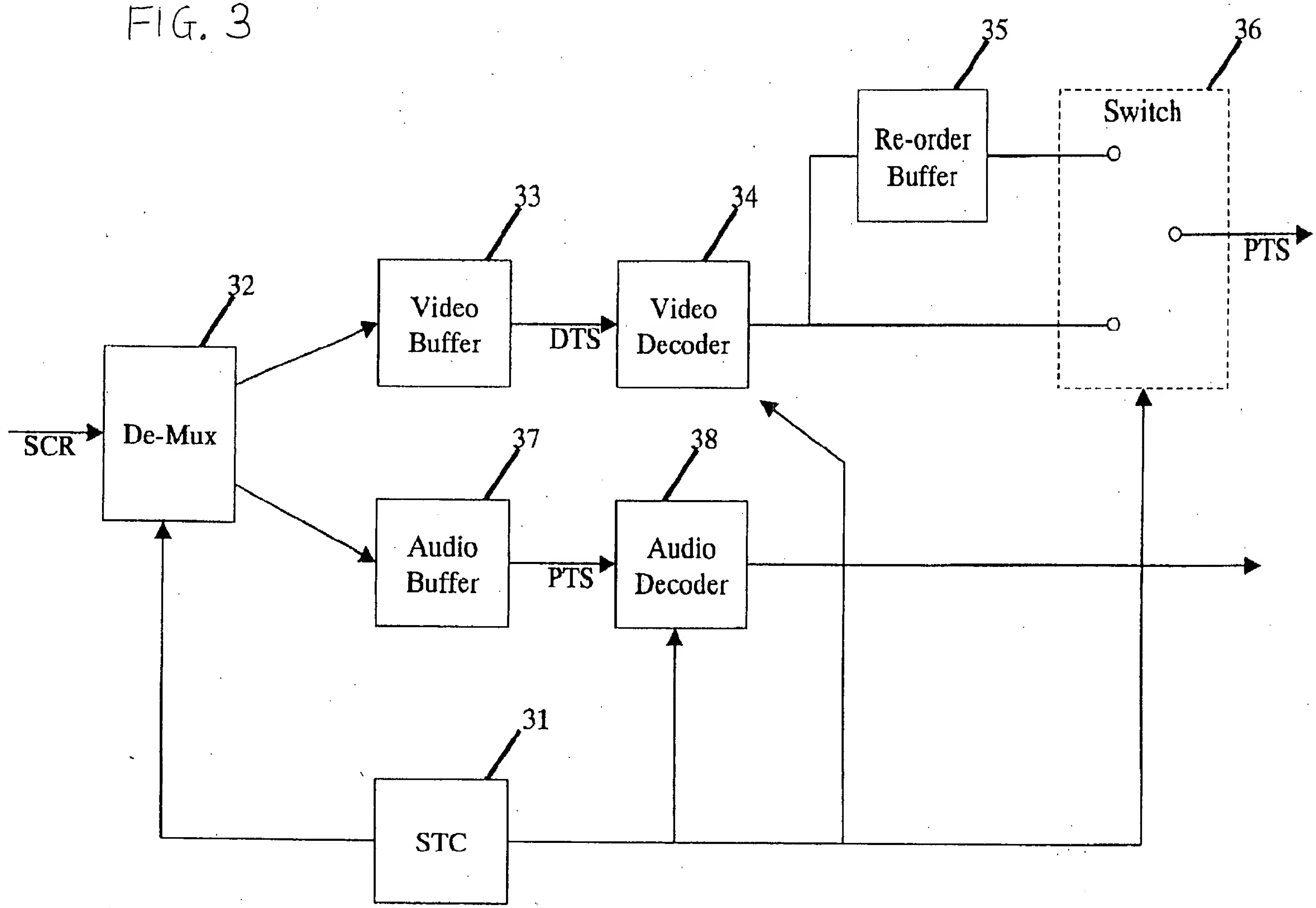
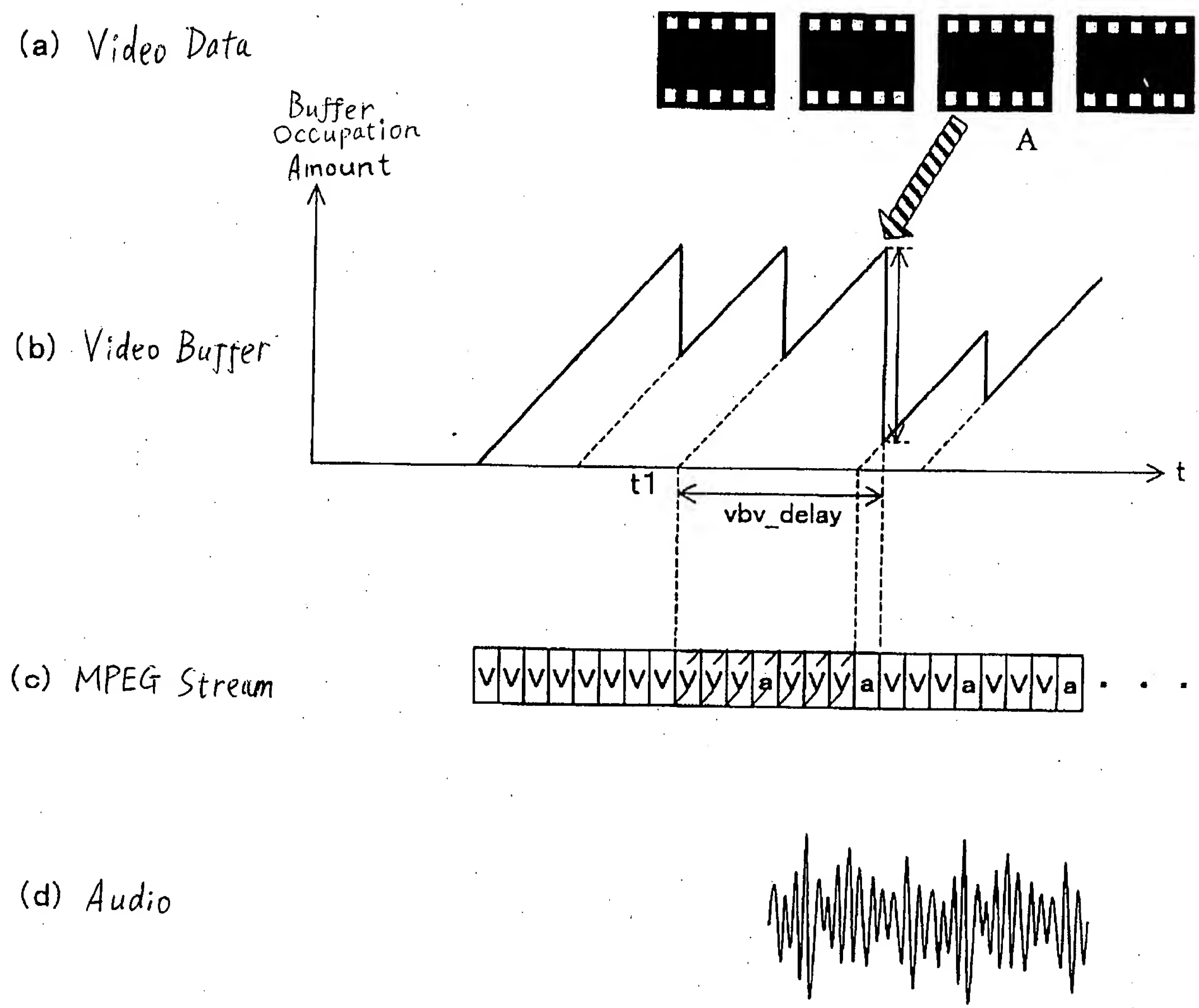

Elementary Stream
21
22
23
Pack Header
Packet Header
Payload
Sector
2KB

FIG. 3



32
De-Mux
SCR
33
Video
Buffer
DTS
34
Video
Decoder
35
Re-order
Buffer
36
Switch
PTS
37
Audio
Buffer
PTS
38
Audio
Decoder
31
STC

FIG. 4



(a) Video Data
A
(b) Video Buffer
Buffer Occupation Amount
t
t1
vbv_delay
(c) MPEG Stream
v v v v v v v v v' v' v' a v' v' v' a v v v a v v v a . . .
(d) Audio

FIG. 5

(a)

(b)

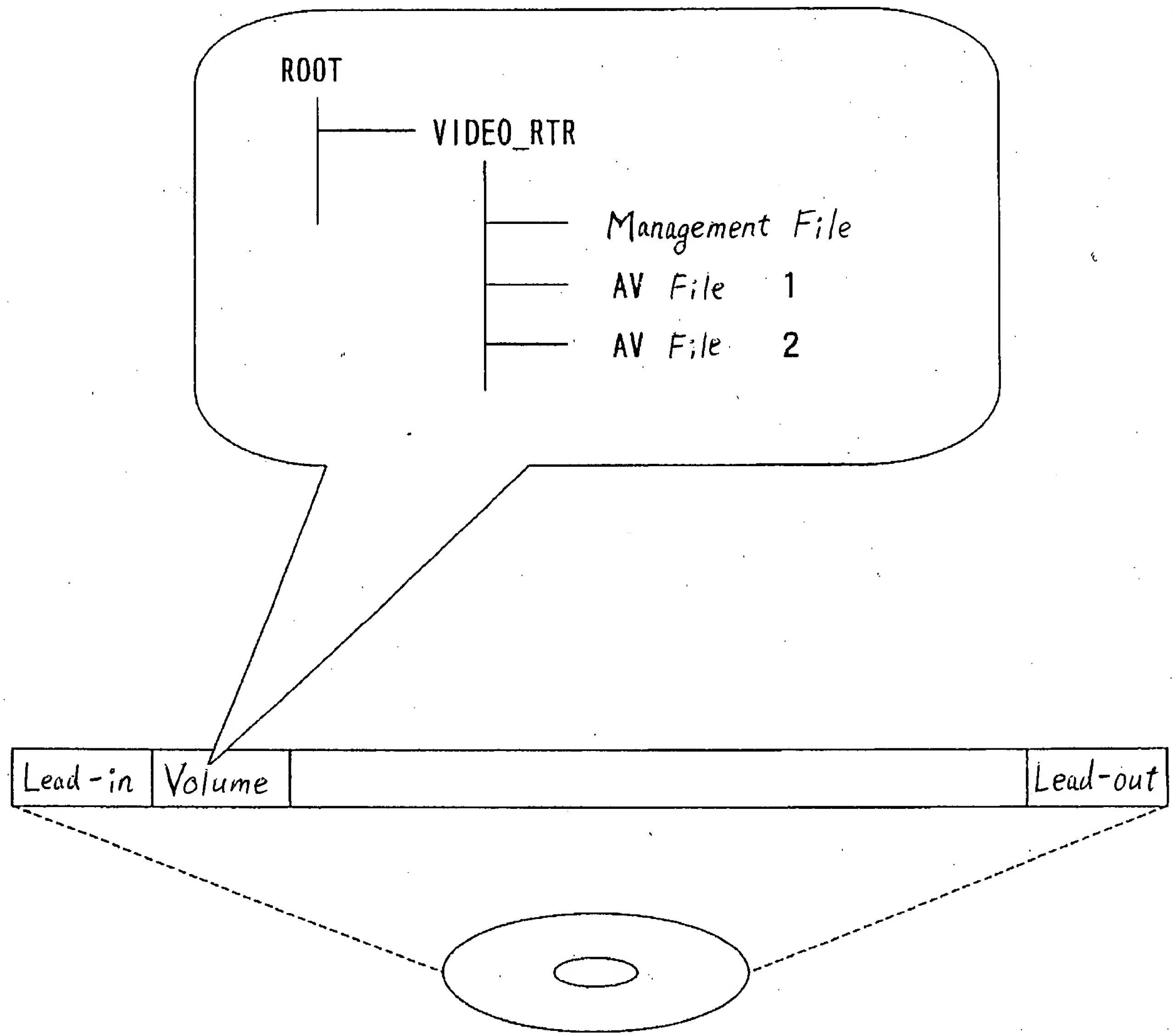
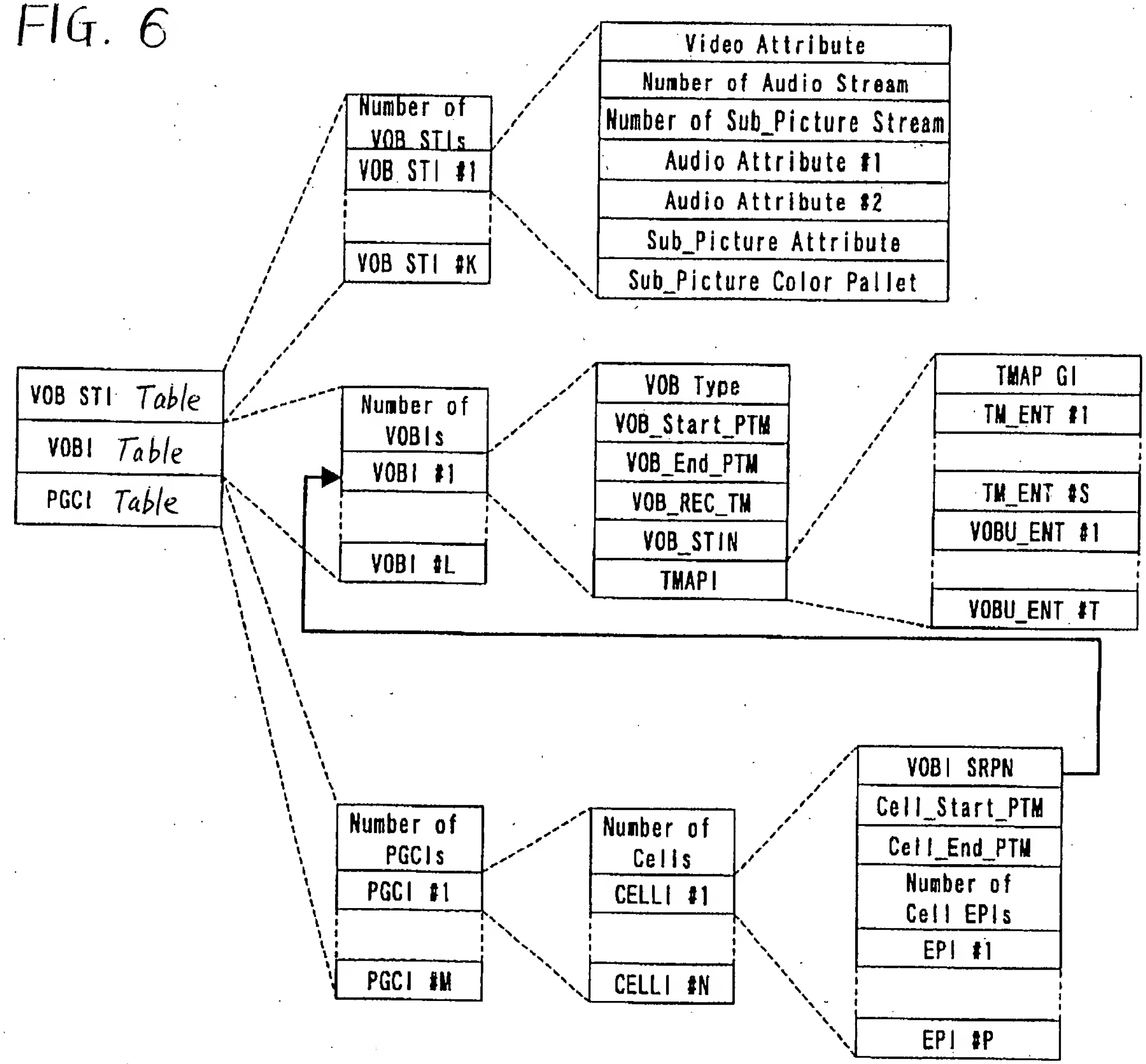
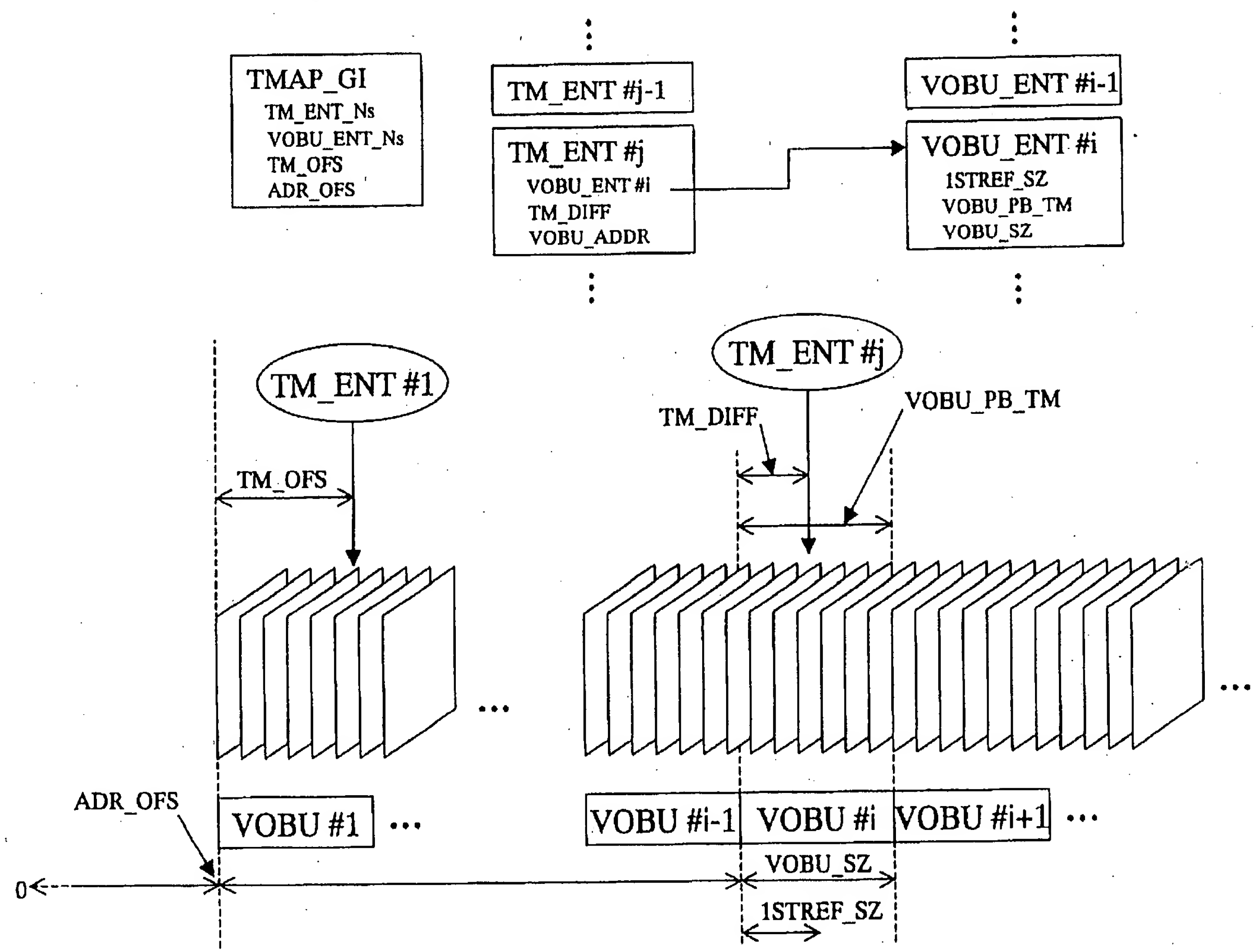

ROOT
VIDEO_RTR
Management File
AV File 1
AV File 2
Lead-in
Volume
Lead-out

FIG. 6



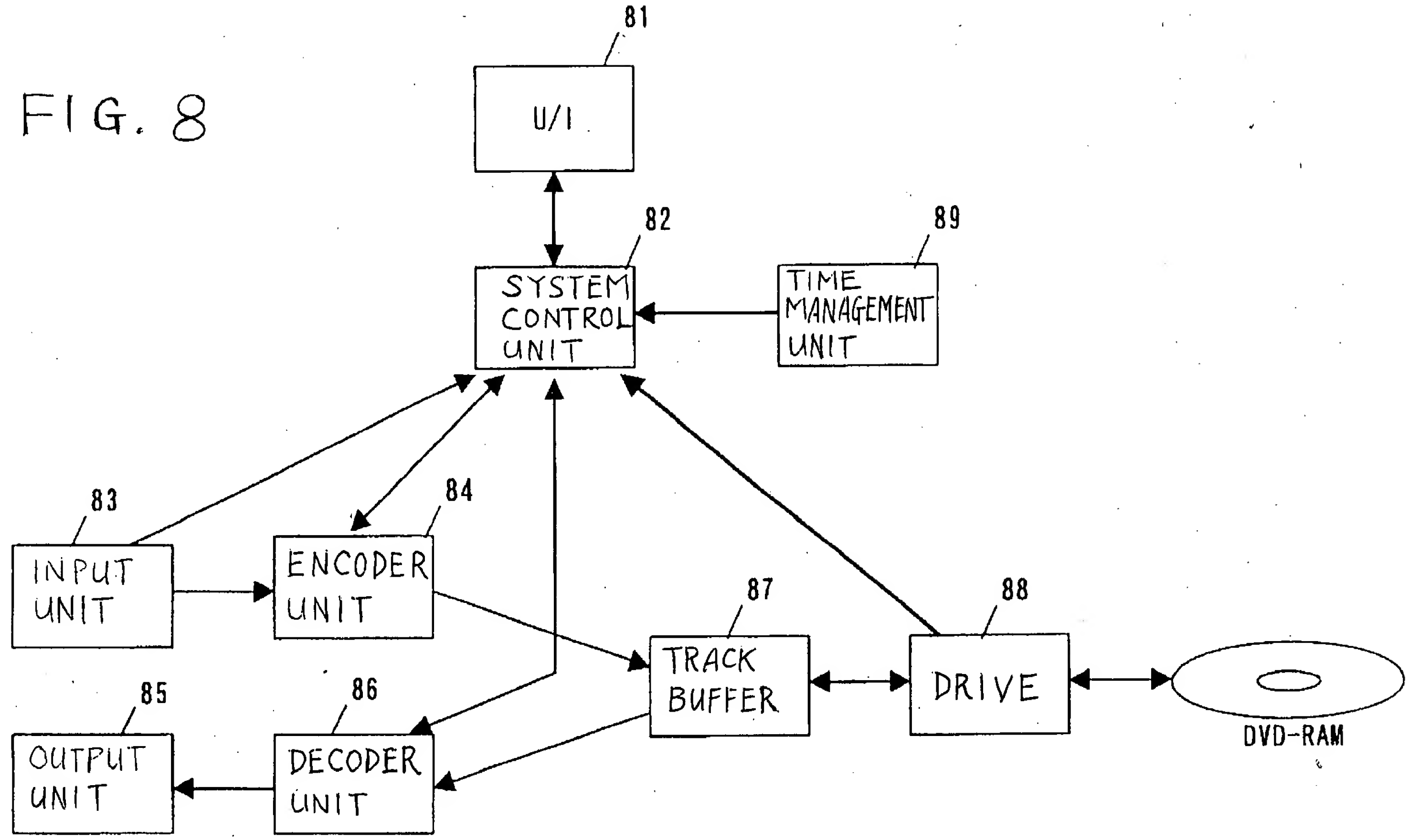
VOB STI Table
VOBI Table
PGCI Table
Number of VOB STIs
VOB STI #1
VOB STI #K
Video Attribute
Number of Audio Stream
Number of Sub_Picture Stream
Audio Attribute #1
Audio Attribute #2
Sub_Picture Attribute
Sub_Picture Color Pallet
Number of VOBIs
VOBI #1
VOBI #L
VOB Type
VOB_Start_PTM
VOB_End_PTM
VOB_REC_TM
VOB_STIN
TMAPI
TMAP GI
TM_ENT #1
TM_ENT #S
VOBU_ENT #1
VOBU_ENT #T
Number of PGCIs
PGCI #1
PGCI #M
Number of Cells
CELLI #1
CELLI #N
VOBI SRPN
Cell_Start_PTM
Cell_End_PTM
Number of Cell EPIs
EPI #1
EPI #P

FIG. 7



TMAP_GI
TM_ENT_Ns
VOBU_ENT_Ns
TM_OFS
ADR_OFS
TM_ENT #j-1
TM_ENT #j
VOBU_ENT #i
TM_DIFF
VOBU_ADDR
VOBU_ENT #i-1
VOBU_ENT #i
1STREF_SZ
VOBU_PB_TM
VOBU_SZ
TM_ENT #1
TM_OFS
TM_ENT #j
TM_DIFF
VOBU_PB_TM
ADR_OFS
VOBU #1
VOBU #i-1
VOBU #i
VOBU #i+1
VOBU_SZ
1STREF_SZ
0

FIG. 8



81
U/I
82
SYSTEM CONTROL UNIT
89
TIME MANAGEMENT UNIT
83
INPUT UNIT
84
ENCODER UNIT
87
TRACK BUFFER
88
DRIVE
DVD-RAM
85
OUTPUT UNIT
86
DECODER UNIT

FIG.9

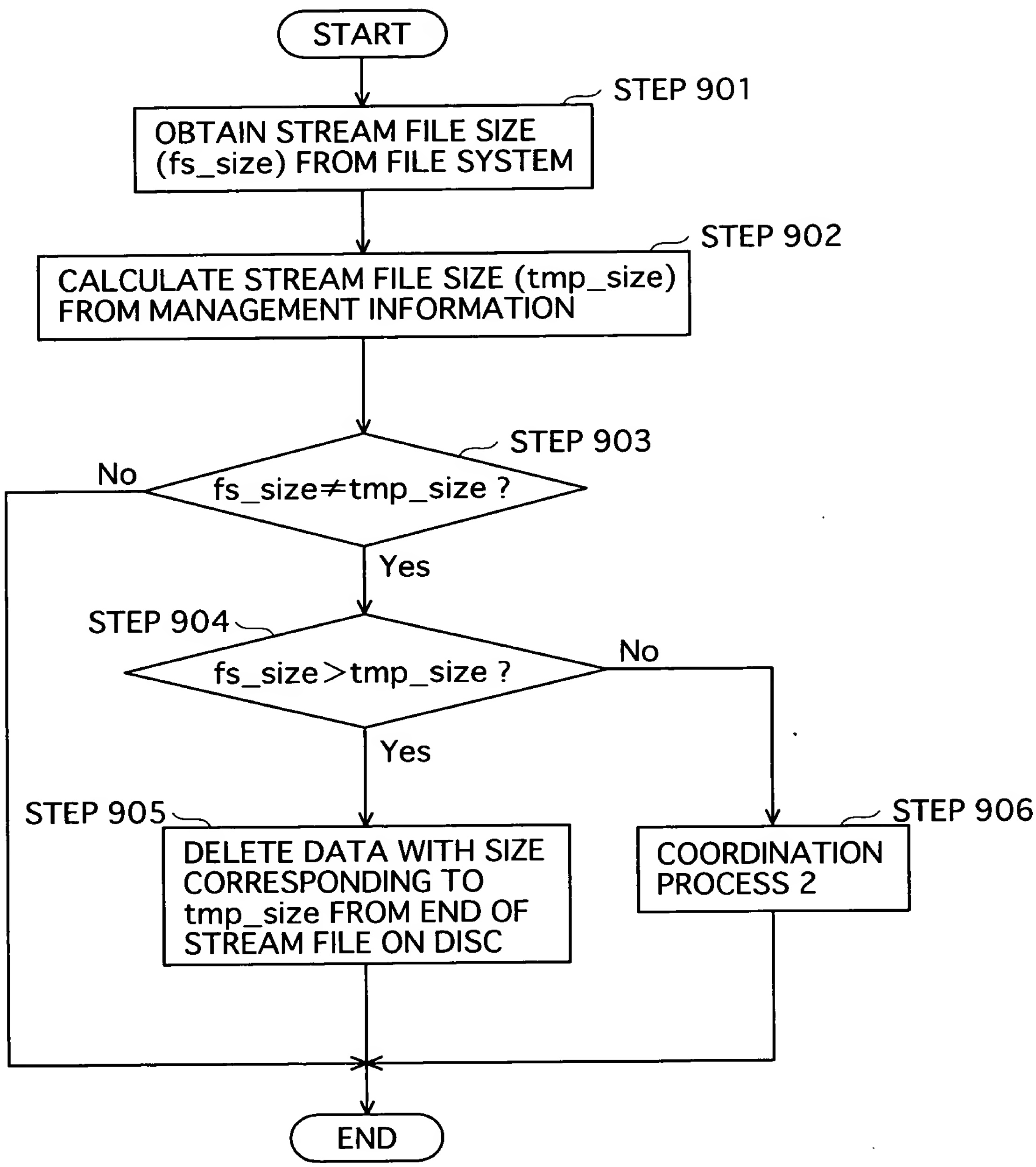
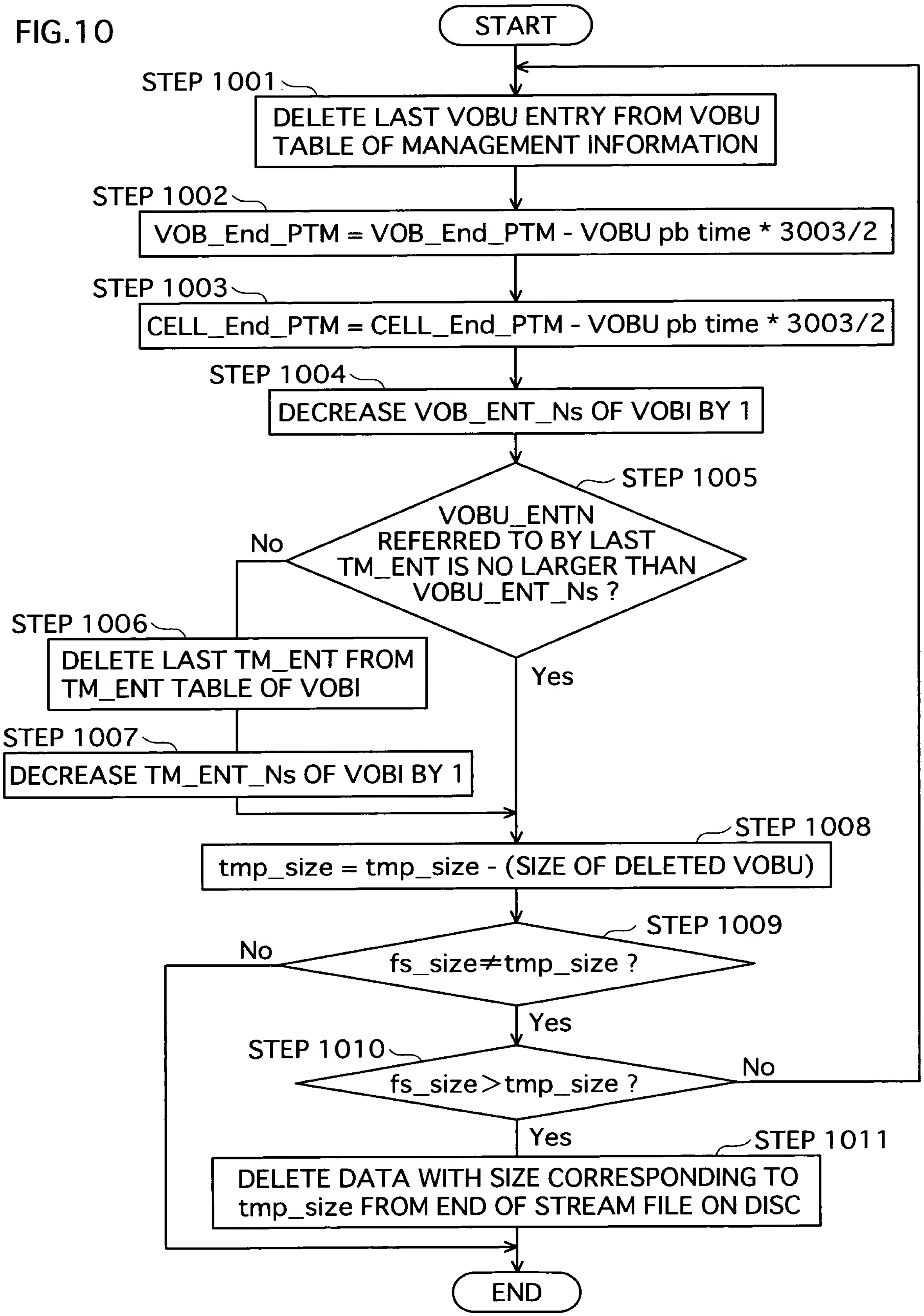

START
STEP 901
OBTAIN STREAM FILE SIZE (fs_size) FROM FILE SYSTEM
STEP 902
CALCULATE STREAM FILE SIZE (tmp_size) FROM MANAGEMENT INFORMATION
STEP 903
fs_size≠tmp_size ?
No
Yes
STEP 904
fs_size>tmp_size ?
No
Yes
STEP 905
DELETE DATA WITH SIZE CORRESPONDING TO tmp_size FROM END OF STREAM FILE ON DISC
STEP 906
COORDINATION PROCESS 2
END

FIG.10


START
STEP 1001
DELETE LAST VOBU ENTRY FROM VOBU TABLE OF MANAGEMENT INFORMATION
STEP 1002
VOB_End_PTM = VOB_End_PTM - VOBU pb time * 3003/2
STEP 1003
CELL_End_PTM = CELL_End_PTM - VOBU pb time * 3003/2
STEP 1004
DECREASE VOB_ENT_Ns OF VOBI BY 1
STEP 1005
VOBU_ENTN REFERRED TO BY LAST TM_ENT IS NO LARGER THAN VOBU_ENT_Ns ?
No
Yes
STEP 1006
DELETE LAST TM_ENT FROM TM_ENT TABLE OF VOBI
STEP 1007
DECREASE TM_ENT_Ns OF VOBI BY 1
STEP 1008
tmp_size = tmp_size - (SIZE OF DELETED VOBU)
STEP 1009
fs_size≠tmp_size ?
No
Yes
STEP 1010
fs_size>tmp_size ?
No
Yes
STEP 1011
DELETE DATA WITH SIZE CORRESPONDING TO tmp_size FROM END OF STREAM FILE ON DISC
END